****

# 《生产实习》

## 2020-2021学年第三学期课程考核报告

### 题 目：运用深度强化学习方法DQN完成

### Atari游戏breakout的AI设计

### 姓 名：

### 学 号：

### 班 级：

### 成 绩：

### 主讲教师签字：

2021年07月23日

1. **生产实习题目**

运用深度强化学习方法DQN完成Atari游戏breakout的AI设计

1. **实习内容及应用场景**

实习内容**：**Atari游戏AI设计：

（1）理解深度学习基础概念和强化学习基础概念；

（2）熟悉gym[atari]环境中的breakout游戏，学会控制游戏的进行，了解设计AI需要处理的基本问题；

（3）学习DQN算法，利用python语言设计AI；

（4）针对breakout游戏进行训练。

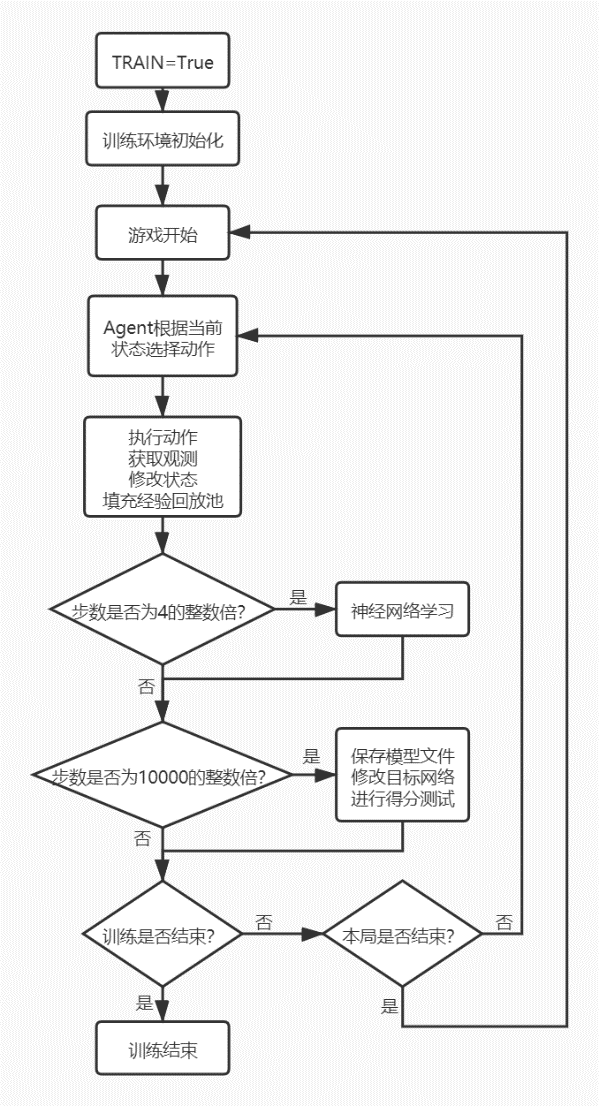
应用场景：Atari中的breakout游戏

1. **小组成员及分工**

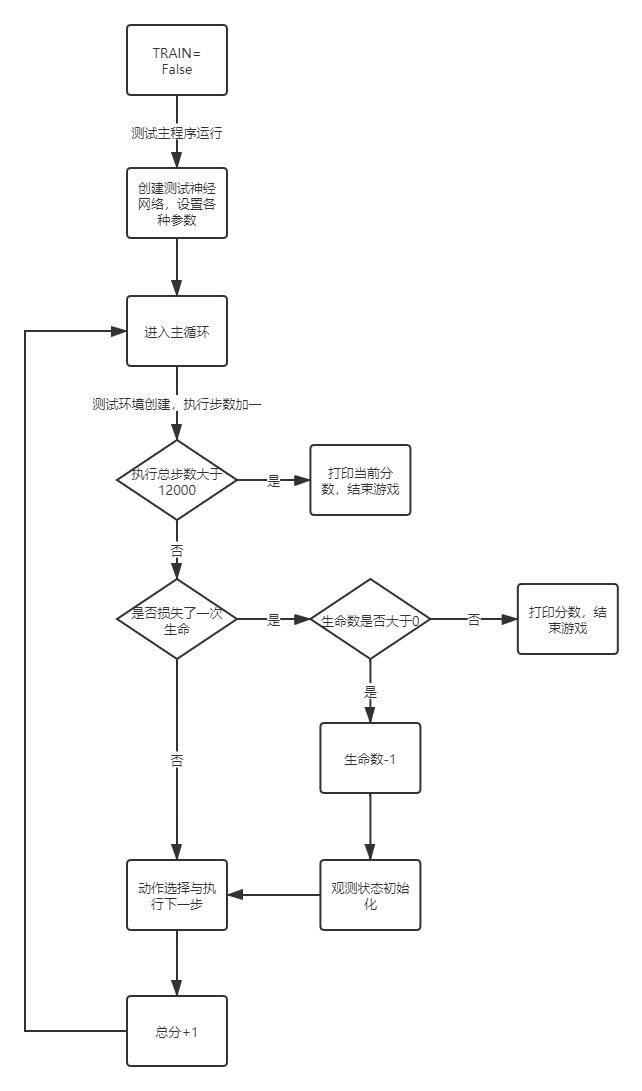
wg：训练与实际运行主程序的编写，训练过程中测试得分程序与环境的编写，对各种参数的调试与模型训练，查阅相关资料进行程序改进与纠错。lym：负责代码部分Agent的设计工作，参与了神经网络结构和训练主循环部分的检查排错工作，辅助完成了展示部分的代码，参与了对各种参数的调试和模型训练工作。

lt：主要负责训练模型所需的神经网络框架编写；对输入图像做裁剪，分割，转换等处理，利于模型参数收敛；参与agent部分代码编写，调试模型参数并对其进行优化。

1. **整体方案设计**
2. 训练运行方案流程设计

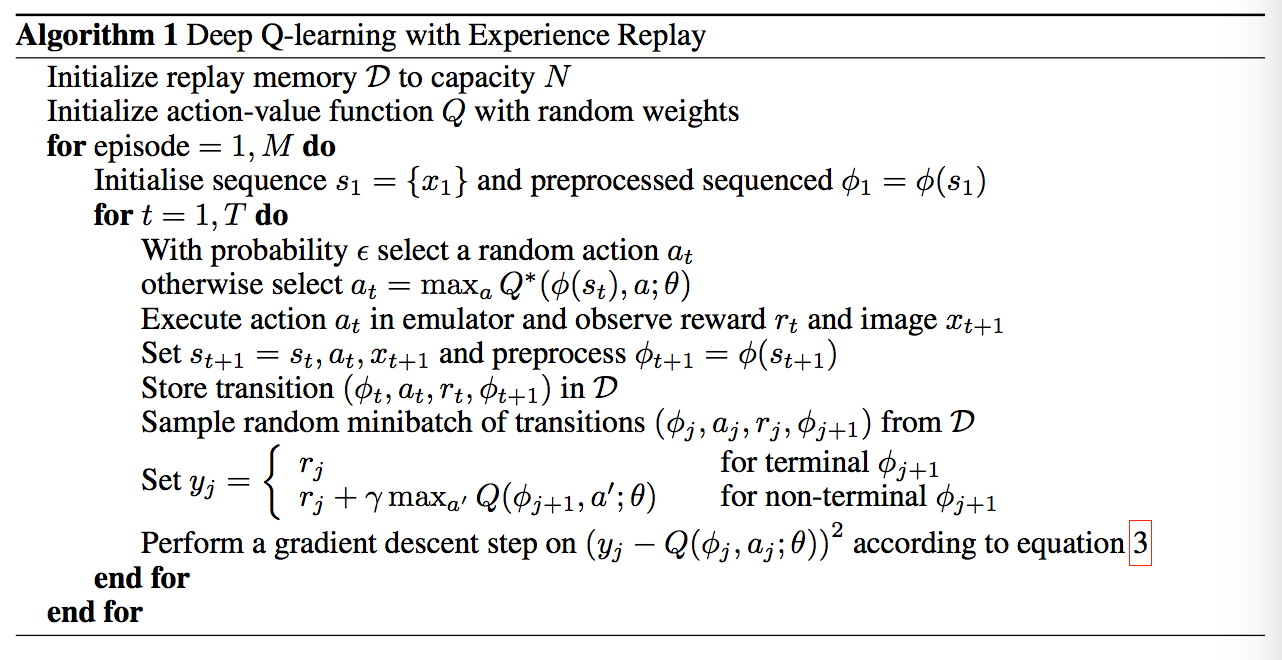


2、测试运行方案流程设计



1. **核心算法**

**DQN算法：**

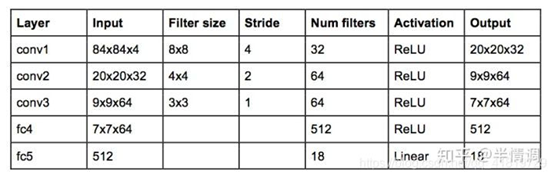
**DQN的由来和解释：**

Q-Learning可以很好的解决迷宫问题，但这终究是个小问题，它的状态空间和动作空间都很小。而在实际的情况下，大部分问题都有巨大的状态空间或动作空间，建立Q表，内存是不允许的，而且数据量和时间开销也是个问题。

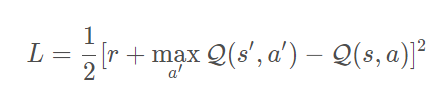
我们可以使用神经网络来表示我们的 Q 函数，每层网络的权重就是对应的值函数，取 4 四帧游戏图像作为 state，输出每个 action 对应的 Q 值。如果我们想要执行 Q 值的更新，或者选择具有最高 Q 值的对应的 action，我们只需经过整个网络一次就能立刻获得任意动作对应的 Q 值。DQN的原始输入为连续的4帧图像，不只使用一帧画面是为了感知环境的动态性。

假设一帧图像有84个像素点，那么四帧图像就有84\*84\*4个像素。每一个像素点的取值是0-255可能，所有的可能性就有256^(84\*84\*4)。

DeepMind使用的网络结构如下：



这是一个经典的带有三层卷积层的卷积神经网络，后面跟两个全连接层。注意：这里没有池化层，池化层会让你获得平移不变性，即网络对图像中对象的位置变得不敏感。这对于 ImageNet 这样的分类任务来说是有意义的，但游戏中位置对潜在的奖励至关重要，我们不希望丢失这些信息。

这个网络的输入是4个84 x 84 的灰度游戏屏幕，输出是每个可能的动作对应的 Q 值 （DeepMind 实验玩的游戏是 Atari，对应有 18 种动作）。这成为一个回归任务，可以用简单的平方误差损失进行优化：  
 

对于给定的<s, a, r, s'>，前一算法的 Q表的更新规则应该做如下修改：

1.对当前状态 s 进行一次前馈，获得所有 action 预测的 Q 值。

2.对下一个状态 s’ 进行一次前馈，计算整个网络最大的输出值 max Q

3.为动作设置目标Q值 r + max Q

4.通过反向传播更新权重。

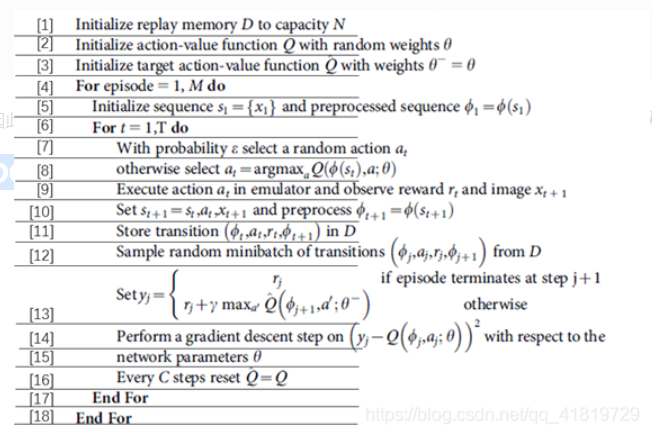
DQN对Q-Learning的修改主要体现在三个方面：

1.DQN利用深度卷积神经网络逼近值函数；

2.DQN利用经验回放（experience replay）训练强化学习的学习过程；

3.DQN独立设置了目标网络来单独处理时间差分算法中的TD偏差

**DQN的伪代码：**

****

1. 初始化回放记忆D，可容纳的数据条数为N
2. 利用随机权值θ来初始化动作-行为值函数Q
3. 令θ-=θ初始化用来计算TD目标的动作行为值Q
4. 循环每次事件
5. 初始化事件的第一个状态s1，预处理得到状态对应的特征输入
6. 循环每个事件的每一步
7. 利用概率ε选一个随机动作at
8. 如果小概率事件没发生，则用贪婪策略选择当前值函数最大的那个动作：
9. 在仿真器中执行动作at
10. 更新st+1=st，设置at，xt+1,预处理
11. 将转换结果储存在回放记忆D中
12. 从回放记忆D中均匀随机采样一个转换样本数据，用来表示
13. 判断是否是一个事件的终止状态，若是终止状态则TD的回报为rj,否则利用TD目标网络参数θ-计算TD回报
14. 执行一次梯度下降算法
15. 更新动作值函数逼近的网络参数θ=θ+Δθ
16. 每隔C步更新一次TD目标网络权值即令θ-=θ
17. 结束每次事件内循环
18. 结束事件间的循环
19. **游戏测试结果**

1、游戏测试结果如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

2、结果分析：

1）程序大部分情况下的游戏运行结果得分为400分左右，即大平均成绩，如图一。

2）程序的最好成绩一般都是差一个方块没有打掉，但是每次差的一个砖块位置并不一样，就导致最后得分也不一样，目前最高分为428分，图二为剩一个砖块得分为425分的情况。

3）因为在砖块还剩较少的情况下，程序容易进入一个一直打不到的死循环，为了避免这种情况，我们设置了一个最大执行步数，经过多次实验后发现12000步比较合适，到达最大执行步数后会打印当前得分并直接退出游戏，即图三情况，命数未用完。

1. **遇到的问题及解决办法**

问题1、对DQN算法不理解，导致代码写不出来

解决方法：在网上查资料，加深对DQN算法的理解，掌握DQN算法的编程流程。

问题2、对gym库中breakout游戏环境框架不了解

解决方法：在网上查资料，看gym库中Atari游戏的环境框架和重要函数和参数。

1. **心得体会**

本次生产实习做的是运用深度强化学习方法DQN完成Atari游戏breakout的AI设计；在这学期，我们学的机器学习课程与本次的生产实习有一定的相关性。本次生产实习除了需要了解DQN算法的原理之外，还需要拥有一定的编程基础，知道如何配置程序运行环境。了解DQN算法原理需要上网看文献，查资料，配置环境则按学长给的教程来操作。在实习过程中，遇到过很多的困难和挫折，例如：刚开始编程时的无从下手；配置环境总是报错时的艰辛；程序写完反复检查无误后却运行不了的迷茫。但是只要我们不放弃，遇到问题多查，多问，问题总会解决。例如，在配置环境的时候我的电脑总是报错，查了很久也没有解决，最后是问了学长，在学长的帮助下解决了；在编程时无从下手时，在网上查找类似代码，仿照别人的模式来编程。经过这次生产实习，无疑大大提高了我的学习能力和解决问题的能力；培养了我不惧困难的品质；收获颇丰！

学生本人签字：

日期：2021年 7 日 22 日

**主讲教师评分及评语：**